Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 61 (2015) 141 – 146

Procedia Computer Science

Complex Adaptive Systems, Publication 5
Cihan H. Dagli, Editor in Chief
Conference Organized by Missouri University of Science and Technology
2015-San Jose, CA

Using Semantic Web Technologies for Integrating Domain Specific Modeling and Analytical Tools

Mark R. Blackburn[a]*, Peter O. Denno[b]

[a]*Stevens Institute of Technology, Castle Point on Hudson, Hoboken, NJ, 07030, USA*
[b]*National Institute of Standards and Technology, 100 Bureau Drive, Gaithersburg, MD, 20899-8265, USA*

Abstract

This paper discusses the potential advantages and pitfalls of using semantic web technologies for representing and integrating modeling and analysis tools. Analytical tools are often not designed to be integrated with information sources and general-purpose modeling tools, and often do not support detection of problems across domains. Additionally, these modeling tools may not capture and represent explicitly the information needed to leverage the capabilities of analysis tools. The method described uses semantic web technology as the integrating mechanism between domain specific modeling (DSM) tools and analytical tools. We describe a method and tool set for representing the analytical knowledge through semantic web ontologies that map between the metamodels of both the DSM and analytical tools. We compare an earlier tool-chain prototype with a significantly revised prototype to reflect on the benefits from using semantic web technologies as an integrating mechanism. A potential advantage is the ability to represent the relationships between modeling and analytical tools explicitly and transparently.

Peer-review under responsibility of scientific committee of Missouri University of Science and Technology

Keywords: domain specific modeling; cyber physical systems; metamodeling; ontologies; semantic web; model-centric engineering;

1. Introduction

The pervasive use of networking, sensors and information technologies to create smart or intelligent systems offers increased effectiveness, productivity, and safety; and they enable increased functionality in smart manufacturing – and, more generally, Cyber Physical Systems (CPS). Model-centric engineering (MCE) is increasing in use to deal with the increased complexity in analyzing both the problem and solutions for CPS. MCE is

* Corresponding author. Tel.: 1-561-637-3452.
E-mail address: mark.blackburn@stevens.edu

Peer-review under responsibility of scientific committee of Missouri University of Science and Technology
doi:10.1016/j.procs.2015.09.174

an overarching digital approach for integrating different model types and tools for simulations and analysis of systems and components at different levels of abstraction and fidelity across disciplines throughout the lifecycle[1]. MCE technologies enable more automation and efficiencies; however, there is still a lack of cross-domain model interoperability and consistency. Additionally, there are limitations transforming models with the required semantic precision, which is needed to provide accurate information for some required analysis.

In the context of this paper, we are concerned with disparate modeling viewpoints and the associated analytical knowledge that is required to leverage analysis tools to support decision making about integration of both computationally enabled equipment and functions of a smart manufacturing system. These viewpoints are modeled using DSMs that can require cross-domain analyses for detecting anomalies and incompatibilities that may arise when new capabilities are introduced into systems. We extend a prototype used in prior research[2] because that work documents the analytical results and benefits of integrating DSM tools with analysis tools†. The prototype discussed herein keeps fixed the DSM metamodel, application models and analysis tool capabilities in order to compare and contrast the use of semantic web technologies as an analytical knowledge representation and integrating mechanism. We focus on declarative and traceable means to compose the information needed to run analytical models.

In the fields of design and engineering, knowledge can be classified along several dimensions: formal versus tacit, product versus process, and, compiled versus dynamic[3]. Sowa describes knowledge representation as a multidisciplinary subject that combines techniques from logic, ontology, and computation[4]. Ontologies represent a possible way to generate a more flexible data model integrating disparate knowledge domains[5]. We want to make various aspects of these domains explicit by formalizing unstructured and tacit knowledge. We believe logic, ontology, and computation are key aspects for formalizing different types of knowledge for cross-domain and multidisciplinary analyses.

The semantic web technologies are based on a standard suite of languages, models, and tools that are suited to knowledge representation. Fig. 1 provides a perspective on the semantic-web-technology stack, which includes eXtended Markup Language (XML)[6], Resource Description Framework (RDF)[7] and Schema (RDFS)[8], Web Ontology Language (OWL)[9], querying language (SPARQL)[10], and others. RDF can describe instances of ontologies. RDFS extends RDF and provides primitives such as Class, subClassOf, and subPropertyOf. The semantic web technologies were created to extend the current Internet by allowing combinations of metadata, structure, and various technologies that enable machines to derive meaning from information, thereby assisting and reducing human intervention. This technology is generally applicable beyond its original intent, as we discussed in this paper.

The technology layers of the semantic web support different levels of abstractions. OWL has found acceptance as a standard notation for knowledge representation. OWL-enabled modeling tools are available from multiple providers, as are supporting assets such as reasoners and application-programming-interface libraries, etc. OWL has been applied to diverse projects in a wide array of fields[11]. OWL was developed, from the beginning, based on formal logical principles; as such, it provides strong support for verification of consistency and satisfiability, extraction of entailments, and conjunctive query answering. This emphasis on formal logic counterbalances the absence of any graphical notation conventions in the OWL standards[12]. Some researchers have attempted to use RDF for related model-based analyses effort such as requirement representation and trade space analysis[11]; however, these attempts have required using code in the transformation to perform needed functions such as inferencing, which is supported more directly by OWL-capable tools.

We focus on the use of the semantic web technologies at the ontology and reasoning layers to represent analytical knowledge as reflected in Fig. 1. The notion of a metamodel of a DSM is strongly related to the notion of domain ontology[13], because each is an abstraction of a conceptualization[14]. Ontologies in OWL are associated with metamodels of a DSM and they also map to the metamodels of analysis tools. Our interest is in the methods of representing the analytical knowledge to take advantage of a standards-based approach to knowledge representation, transformation, and formal analysis. The transformations are applied to specific instances of information derived from models as subject, predicate and object using RDF triples that are compliant with their respective ontology.

---

† Certain commercial software products are identified in this paper. These products were used only for demonstration purposes. This use does not imply approval or endorsement by NIST, nor does it imply these products are necessarily the best available for the purpose. □

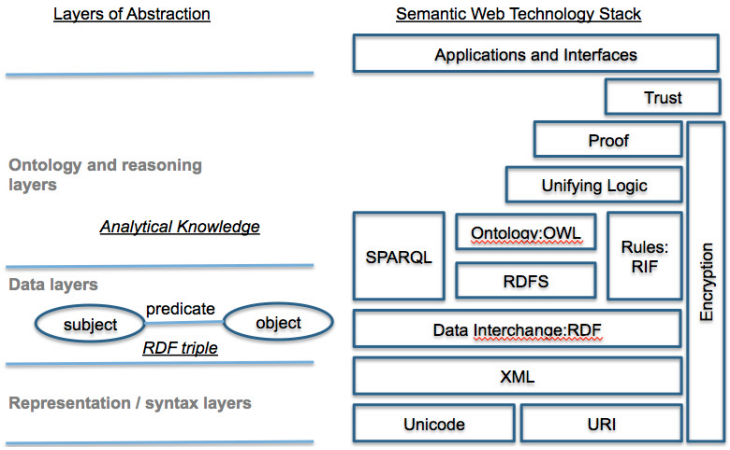

Fig. 1. Semantic Web Technologies related to Layers of Abstraction.

2. Context

There are many stakeholders involved in various roles that contribute to both the problem formulation and solution of a CPS. These stakeholders have differing concerns related to trade space, design, integration, safety, and operations, among others. A viewpoint establishes the purpose and audience for a representation of a system[15]. Viewpoints relevant to production include representations of schedules, process plans, inspection results, inventory, unit process descriptions, and equipment datasheets. Each of them has some type of conceptualization of how their view applies to the overall problem formulation and process as reflected in Fig. 2. The conceptualizations of these views can be represented in a semantically precise way using ontologies[16].

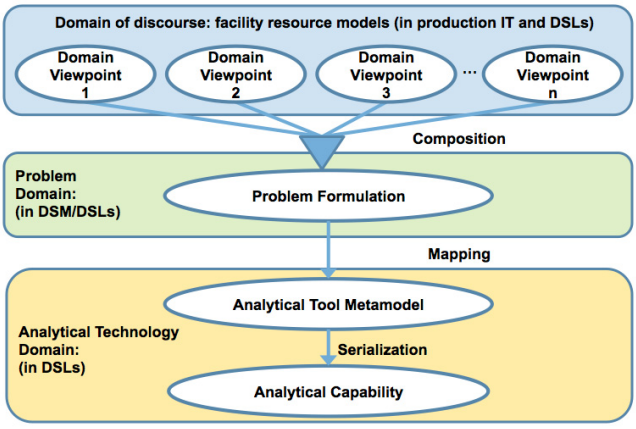

Fig. 2. Transforming Views of the Problem in order to Leverage Analytical Capabilities for a Specific Objective.

By expressing the problem-formulation metamodel as an OWL ontology, we enable three capabilities that are not easily achieved by other means. First, the problem-formulation ontology represents a composition of more fundamental viewpoints that are oftentimes provided without interrelation in the domain of discourse. The problem-formulation ontology enables an analysis of the compositionality of those views. For example, the interconnection of component equipment may be provided by a piping and instrumentation diagram (P&ID), whereas a behavioral viewpoint is provided by a mechanical control view. It is possible that some entailments of the composition of

viewpoints are incompatible. For example, the controls view may suggest that a component valve provides a regulating function, whereas the equipment view indicates that the same valve is only intended to provide safety-pressure relief.

Second, OWL axioms may be applied against the problem-formulation ontology to ensure that individual viewpoints are well-formed. For example, pipes, as elements of the P&ID view, may be constrained to have connections on two ends. This example illustrates one of the shortcomings in our first prototype[2]; the DSM system captured the needs for these types of constraints, however, they were not visible to the analytical tool through the model transformation.

Third, the problem-formulation ontology enables traces to requirements. Tracing requirements is problematic because 1) design tools oftentimes do not enable annotation of requirements on design elements or 2) such annotations are only possible on abstract system viewpoints, such as those provided by SysML[1]. The openness of ontology-based development enables requirements to be superimposed, in an ad hoc manner, over existing viewpoints.

The use of an ontology for the problem formulation does, however, presents its own challenges. Foremost among them is that it does not provide a straightforward means to organize mappings to the analytical-tool metamodel. As suggested in our earlier work,[18] analytical metamodels emphasize structural containment and part-whole relationships because it is commonplace for software to require syntactically structured input.

3. Objective and Approach

We significantly modified the tool chain used in prior efforts[2], as shown in Fig. 3. The context of that research focused on virtual design and verification of industrial process-plants' designs. The prototype used DSMs and DSLs of a system design, provided examples of how the integration with formal methods can identify defects in the design, and automatically generates test vectors with requirement-to-test traceability. The project research involved three main roles: 1) developing the DSM metamodel for integrated system designs, 2) creating application-specific models, using two graphical DSLs, and 3) producing the generator required to demonstrate analysis and test generation. The elements of the prior prototype are shown as shaded elements such as the DSM (MetaEdit+[19]) metamodels, associated application models and T-VEC[20] analysis, test vector generation and requirement-traceability capabilities. These elements remain fixed in the updated prototype in order to formulate a comparison of the potential advantages and pitfalls of the new approach over the prior approach.

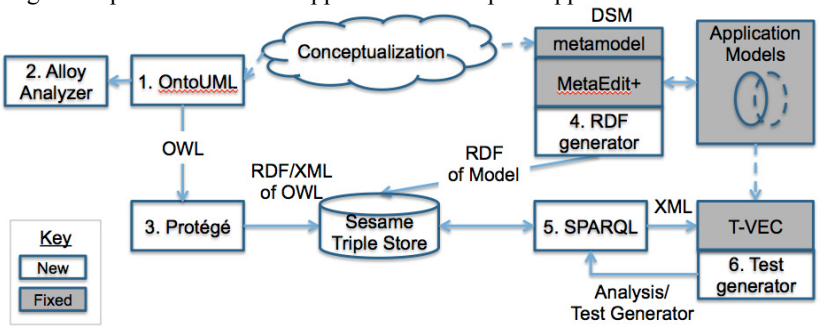

Fig. 3. Ontology, Domain Specific Modeling, Analysis and Semantic Web Prototype

As detailed below, the modifications to the prototype are reflected in the unshaded elements in Fig. 3. These elements provide the new functionality of the prototype. They are used to 1) model and analyze ontologies used by the semantic web technologies and 2) represent the analytical knowledge needed to transform model instances into the representations required by the analysis tools (i.e., Alloy Analyzer[21], T-VEC). The functions performed by the new elements in Fig. 3 are labeled below.

1. We created the initial ontology with Protégé, an open-source ontology editor and framework[22], but adopted a more rigorous method for developing a domain ontology representation, in which we used a lightweight modeling

languages and tool, OntoUML lightweight editor (OLED)[23]. We modeled our conceptualization of the P&ID domain using OLED, which also produces an OWL ontology in XML. Having a precise representation of a given conceptualization becomes even more valuable when we want to integrate different, independently developed models or systems based on those models[24] as reflected by the problem formulation discussed in Section 2.

2. OLED also produces an Alloy specification[21]. The Alloy Analyzer is a solver that checks an Alloy specification for well-formedness properties and satisfiability of constraints. The Alloy Analyzer also produces visualizations of possible instances derived from the modeled ontology. This has the added benefit of providing both the ontology modeler and the domain-subject-matter experts 1) a way to visualize possible instances from the ontoUML model and 2) a type of early model validation of the modeled ontology.

3. We use Protégé to convert OWL to RDF/XML for loading into an inferencing-enabled Sesame triple store.

MetaEdit+ is a tool that provides capabilities to represent a conceptualization as a metamodel and allows users to construct specific application models (instances) that are compliant with the metamodel. MetaEdit+ provides a template-based-generator capability to support transformations of application models into artifacts such as documents, code, or, other types of language generation. In our prior effort[2], we used the generator and some additional software to perform the application model transformations into T-VEC specifications; that research was focused on integrating DSMs with formal-method tools like T-VEC. T-VEC is a theorem prover that we used to prove that the different types of properties (e.g., flow, pressure) were valid in the application model. T-VEC also generates test vectors with requirement-to-test traceability; these same capabilities are performed in the new version of the tools.

4. In the new prototype, we created a different MetaEdit+ generator (RDF generator) that produces RDF representations of the application models. The generator was significantly simpler (i.e., about one third the code size of the prior version). We can demonstrate that these models are compliant with the OWL ontology through formal logics. These kinds of capabilities are well documented[12] and complement our approach, because the models have the necessary formalization in OWL and RDF.

5. We created SPARQL queries to extract information and serialize it into the XML-based language of T-VEC. This was also straightforward - eliminating the generator code we used in the earlier prototype. The current version uses SPARQL queries through an application programming interface to the triple store, which made the serialization of the XML to T-VEC easier to do. These same SPARQL queries can be executed from a web browser.

6. The outputs of the analysis and test-vector-generation processes are loaded back into the triple store repository. To examine the results, users can use web browsers to perform SPARQL queries directly or through application program libraries in several different languages. Some researchers have created natural-language interfaces to further simplify the interface and raise the level of abstraction for presenting the semantic web information to subject matter experts in the domain[25].

4. Conclusions

We describe the approach to transform DSMs through semantic web technologies, thereby successfully demonstrating a new variant of our tool set. The new approach uses domain conceptualization, both in terms of metamodels and ontologies, to ensure semantic consistency across different representations. This not only improves on the prior approach, but also provides for a more comprehensive and systematic approach for characterizing the domains associated with the problem-formulation-ontology concept. In addition, we believe that the early validation capabilities provided through model satisfiability checking and the visualization of specific model instances using tools like the Allow Analyzer are valuable for subject-matter experts across the related domains.

The new approach significantly simplifies the DSM generator. It does require more effort in developing the ontology, but that has value as described previously. In addition, the ontology, if defined appropriately, can leverage inferencing in the triple store, which can only be done from either RDFS or OWL. The inferencing creates RDF in the triple store for associations (e.g., class, subclass) derived from the model. These types of associates were previously produced through code in the generator of the early prototype, because they are required for the model transformation into T-VEC. In addition, the new serialization using SPARQL, which is also straightforward, because all of the needed information is in the triple store, again requires significantly less code. SPARQL is a relatively simple, open, standard-based language. Its simplicity facilitates verification. These results, derived

through the application of semantic web technologies, reflect well on our desire to have more transparency of the analytical knowledge, in this case focused mostly on the model transformations.

The methodology underlying the ontoUML approach adds methodological rigor, which like any methodology involves learning; but the rigor and associated analysis and visualization tools pay off through model validation. This approach leads also to the need for methodological rigor in the development of the ontology, because users need to understand how the ontology-generation process works (e.g., the generation of namespaces in the RDF).

The new aspects of the approach are open and standards-based; this addresses some of the needs for semantically precise representation of MCE. Equally important is that the transformation medium is potentially tool agnostic, which can have significant potential benefits when coordinating efforts between companies that don't use the same tooling. A tool-agnostic approach is desirable in the acquisition of complex systems. As we move into a world where we need to share more information digitally, imposing any particular set of tools on the contractors and developers of these systems is problematic. Not only is this information important for early conceptualization and design, but the availability for digitally precise and semantically rich information is more important in manufacturing, operations and sustainment.[26]

References

1. Blackburn, M., R. Cloutier, G. Witus, E. Hole, M. Bone, Transforming System Engineering through Model-Centric Engineering, SERC-2014-TR-044-2, January, 2015.
2. Blackburn, M., P. Denno, Virtual Design and Verification of Cyber-physical Systems: Industrial Process Plant Design, Conference on Systems Engineering Research, March, 2014; http://dx.doi.org/10.1016/j.procs.2014.03.006.
3. Chandrasegaran, Senthil K., Karthik Ramani, Ram D. Sriram, Imré Horváth, Alain Bernard, Ramy F. Harik, and Wei Gao. "The Evolution, Challenges, and Future of Knowledge Representation in Product Design Systems." Computer-Aided Design 45, no. 2, February 2013.
4. Sowa J. Knowledge representation and reasoning: logical, philosophical, and computational foundations. Brooks/Cole; 2000.
5. Terkaj, W., Pedrielli, G., & Sacco, M. (2012). Virtual Factory Data Model. 7th International Conference on Formal Ontology in Information Systems (FOIS 2012). Graz: IOS Press.
6. Object Management Group, XML Metadata Interchange (XMI), Version, 2.4.2, April 2014, http://www.omg.org/spec/XMI/2.4.2.
7. RDF – Resource Description Framework (RDF): Concepts and Abstract Syntax, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/rdf-concepts/.
8 World Wide Web Consortium. RDF Vocabulary Description Language 1.0: RDF Schema. http://www. w3.org/TR/rdf-schema/, February 2004.
9. World Wide Web Consortium. OWL 2 Web Ontology Language Document Overview. 2009. Available from: http://www.w3.org/TR/2009/REC-owl2-overview-20091027/.
10. SPIN – SPARQL Inferencing Notation: Overview and Motivation, W3C Member Submission, 22 February 2011, http://www.w3.org/Submission/spin-overview/.
11. Nassar, Nefretiti, and Mark Austin. "Model-Based Systems Engineering Design and Trade-Off Analysis with RDF Graphs." Procedia Computer Science 16 (2013): 216–25. doi:10.1016/j.procs.2013.01.023.
12. J. Steven Jenkins, Nicolas F. Rouquette, Semantically-Rigorous Systems Engineering Modeling Using SysML and OWL, Jet Propulsion Laboratory, California Institute of Technology, 2012.
13. Bézivin, J. On the unification power of models. Software and System Modeling, 4(2):171–188, May 2005.
14. Guizzardi, G. "Ontological Foundations for Structural Conceptual Models", Telematica Instituut Fundamental Research Series no. 15, Universal Press, The Netherlands, 2005, ISBN 90-75176-81-3.
15. ISO/IEC 42010:2007, Systems and Software Engineering -- Architecture Description, 2007.
16. Gruber, Thomas R. "Toward Principles for the Design of Ontologies Used for Knowledge Sharing." Int. J. Hum.-Comput. Stud. 43, no. 5–6 (December 1995): 907–28. doi:10.1006/ijhc.1995.1081.
17. OMG System Modeling Language, See http://www.omgsysml.org.
18. Denno, P. O., D. B. Kim, Integrating views of properties in models of unit manufacturing processes, National Institute of Standards and Technology, December 2014.
19. MetaEdit+ Tool Suite, http://www.metacase.com/.
20. T-Vec Tool Suite, http://www.t-vec.com/.
21. Alloy, http://alloy.mit.edu/alloy/.
22. Protégé, http://protege.stanford.edu/.
23. OntoUML lightweight editor , https://code.google.com/p/ontouml-lightweight-editor/
24. Guizzardi, G. "The Role of Foundational Ontologies for Conceptual Modeling and Domain Ontology Representation," 17–25. IEEE, 2006.5doi:10.1109/DBIS.2006.1678468.
25. Sukys, A., Lina Nemuraite, Bronius Paradauskas, Edvinas Sinkevicius, Transformation Framework for SBVR based Semantic Queries in Business Information Systems, The Second International Conference on Business Intelligence and Technology, 2012.
26. Witherell, Paul, Boonserm Kulvatunyou, and Sudarsan Rachuri. "Towards the Synthesis of Product Knowledge Across the Lifecycle," V012T13A071. ASME, 2013. doi:10.1115/IMECE2013-65220.